



Point processes in software reliability

James Ledoux

► To cite this version:

James Ledoux. Point processes in software reliability. Catherine Huber, Nikolaos Limnios, Mounir Mesbah, Mikhail Nikulin. Mathematical Methods in Survival Analysis, Reliability and Quality of Life, John Wiley and Sons, pp.219-236, 2008, Wiley-ISTE, 9781848210103. hal-00856329

HAL Id: hal-00856329

<https://hal.science/hal-00856329>

Submitted on 30 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Chapter 15

Point Processes in Software Reliability

15.1. Introduction

Most systems are now driven by software applications. It is well-recognized that assessing reliability of software products is a major issue in reliability engineering, particularly in terms of cost. An impressive list of system crashes due to software and their cost is reported in [PHA 00]. Thus, a huge amount of human and financial resources are devoted to the production of dependable systems which is, at the present time, the prerogative of the software engineering community. However, a central question is : does the production meet the objectives? To answer this, mathematical methods for reliability assessment of software systems must be proposed. The failure process of a software differs from that of hardware in the following specific aspects:

- 1) we are primarily concerned with design faults. A *fault* (or bug) refers to the manifestation in the code of a mistake made by the programmer or designer with respect to the specification of the software. Activation of a fault by an input value leads to an incorrect output. Detection of such an event corresponds to an occurrence of a *software failure*;

- 2) the software does not wear out. Its reliability is intended to be improved by corrective maintenance;

- 3) the failure process is highly dependent on the operational profile of the software. Activation of a fault depends on the input data and this fault is activated each time the corresponding input data is selected (as long as the fault is not removed).

Chapter written by James LEDOUX, INSA de Rennes & IRMAR.

The prevalent approach in software reliability modeling is the *black-box* approach, in which only the interactions of the software with the environment are considered. More than 50 such models are listed in [PHA 00]. No real effort has been made to include all models in a single mathematical framework, which should make easier a comparative analysis of models. Decision makers in industry face the lack of clarity in the research activities for failure prediction, and have focused their resources on methods for producing dependable softwares. These methods are essentially based on fault prevention and removing. However, this does not respond to the initial question, that is, assessing if the reliability objectives are met. To the best of my knowledge, Gaudoin and Soler made the first significant attempt to clarify the relationships between various models [GAU 90]. Their idea was to use the self-excited point processes as their mathematical framework. With the help of results in Snyder's book [SNY 91], they classified a large part of the standard models and provided their main statistical properties. This was rediscovered by Singpurwalla and his co-authors in [CHE 97, SIN 99]. In fact, this framework can be included in the martingale approach of point processes which allows us to consider a much wider class of models. This was sketched in [LED 03a] and is formed in the forthcoming book [GAU 07].

In a second approach, called the *white-box* approach, information on the structure of the software are incorporated in the models. Software applications increase in size and complexity, so that a satisfactory control of their behavior can-not be expected from a single black-box view of the system. Extensive numerical experiments have shown that black-box models are suitable for capturing a trend in the dependability metrics, but not for obtaining precise estimates. This is a standard need during the preliminary phases of the life of a software. However, specifically when the software is in operation, it is intended that an architecture-based approach allows analyzing the sensitivity of the dependability of the system with respect to those of its components. To save money, such analysis must be done prior to their effective implementation in the system. Moreover, the usage profile of a component of a system during the operating and testing phases differs significantly. Therefore, the failure process of the system depends greatly on the structure of the execution process. However, it is a matter of fact that the contributions to the architecture-based reliability modeling are rare. This can be explained from certain open issues: what is the architecture of a software? What kind of data and how much data can be collected?

Input values may be considered as arriving to the software randomly. So although software failure is not generated stochastically, it is detected in such a manner. Therefore, this justifies the use of stochastic models of the underlying random process that governs the software failures. Specifically, the failure process is modeled by a point process. In section 15.2, the basic concepts of reliability of repairable systems are recalled. Then, the basic set-up for black-box models is introduced in section 15.3. section 15.4 is devoted to white-box modeling. After introducing a benchmark model given by Littlewood, a general Markovian model is presented. Calibration of its parameters and Poisson approximation in the case of reliability growth are discussed.

15.2. Basic concepts for repairable systems

The *time to failure* for a non-repairable system is a random variable X taking values in $[0, +\infty]$. The probability distribution of X is assumed to have a density f over $\mathbb{R}_+ = [0, +\infty[$. The reliability function is defined by

$$\forall x \geq 0, \quad R(x) = 1 - \int_0^x f(u) du.$$

Note that there exist reliability models in which the system may be considered as failure-free. In this case,

$$\mathbb{P}\{X = +\infty\} = 1 - \int_0^{+\infty} f(x) dx > 0.$$

The *failure rate* or *hazard rate* is the main reliability metric of a non-repairable system:

$$\forall x \geq 0, \quad h(x) = \frac{f(x)}{R(x)} = \frac{f(x)}{1 - \int_0^x f(u) du}. \quad (15.1)$$

It characterizes the probability distribution of X via the exponentiation formula. Function $R(\cdot)$ is not very useful in the context of repairable systems. Thus, we adhere to Ascher-Feingold's definition of the reliability function for repairable systems as a generalization of the residual survival function of non-repairable systems [ASC 84]. Indeed, at time t , the probability that the system is operational up to a specified time should depend on all observed events prior to t (failures, corrections or external factors). All these past events are gathered in the *history* (or filtration) at time t , \mathcal{H}_t . Thus, the reliability of a repairable system is a probability conditional to \mathcal{H}_t .

DEFINITION 15.1.— *At time t , the reliability of the system at horizon $\tau \geq 0$, is the probability that the system is operational on the interval $]t, t + \tau]$ given the history at time t . That is, the **reliability function** at t is the function R_t defined by*

$$\forall \tau \geq 0, \quad R_t(\tau) = \mathbb{P}\{N_{t+\tau} - N_t = 0 \mid \mathcal{H}_t\} = \mathbb{P}\{T_{N_t+1} - t > \tau \mid \mathcal{H}_t\}.$$

Let us comment on the concept of *reliability growth*, i.e. the fact that dependability is improving with time. This is interpreted as follows: when the system is observed at time t_2 , posterior to t_1 , the reliability at t_2 is greater than that evaluated at t_1 whatever the horizon τ : for $t_1 < t_2$,

$$\forall \tau \geq 0, \quad R_{t_1}(\tau) \leq R_{t_2}(\tau).$$

Nevertheless, function R_t is non-increasing as a function of τ .

A software is observed from instant $t_0 = 0$. Failures happen at instants $\{t_n\}_{n \geq 1}$. After each of them, the software is either corrected or not, and then rebooted. The standard models assume the following: 1) the delays to recover a safe state are not taken into account. 2) The failure instants can be identified to the moments of request (i.e. the time to execution is neglected). 3) Correction is immediate. Now, each t_n is the observed value of a random variable T_n and $\{T_n\}_{n \geq 1}$ is a *point process*, that is, a sequence of non-negative random variables, all defined on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$, such that:

- 1) $T_0 = 0$ and $T_1 > 0$,
- 2) $T_n < T_{n+1}$ if $T_n < +\infty$, and $T_{n+1} = T_n$ when $T_n = +\infty$.

Note that

$$\lim_{n \rightarrow +\infty} T_n = +\infty. \quad (15.2)$$

This definition covers the important case in software reliability of models with a finite number N of failure times T_1, \dots, T_N ($T_n = +\infty$ for $n \geq N + 1$). A point process is specified by any of the two following stochastic processes:

- 1) the sequence of inter-failure times, $\{X_n\}_{n \geq 1}$, where $X_{n+1} = T_{n+1} - T_n$ if $T_n < +\infty$, and $+\infty$ when $T_n = +\infty$;
- 2) the *counting process* of failures, $\{N_t\}_{t \geq 0}$, with $N_0 = 0$ and $N_t = \sum_{n=1}^{+\infty} \mathbb{I}_{\{T_n \leq t\}}$. Note that condition (15.2) is equivalent to $\mathbb{P}\{N_t < +\infty\} = 1$ for every t .

Let $\{N_t\}_{t \geq 0}$ be adapted to some history $\mathcal{H} = \{\mathcal{H}_t\}_{t \geq 0}$. It is assumed to be *integrable* to simplify the exposition:

$$\forall t \geq 0, \quad \mathbb{E}[N_t] < +\infty. \quad (15.3)$$

We introduce the concept of stochastic intensity which generalizes the concept of failure rate of non-repairable systems.

DEFINITION 15.2.— *Let $\{\lambda_t\}_{t \geq 0}$ be a non-negative and \mathcal{H} -predictable process. Set*

$$\forall t \geq 0, \quad M_t = N_t - \int_0^t \lambda_s ds.$$

$\{\lambda_t\}_{t \geq 0}$ is called the \mathcal{H} -intensity of $\{N_t\}_{t \geq 0}$ if $\{M_t\}_{t \geq 0}$ is an \mathcal{H} -martingale.

Recall that any \mathcal{H} -adapted process with left-continuous paths is \mathcal{H} -predictable. The interest in an \mathcal{H} -predictable intensity is that it is unique up to a set of $\mathbb{P} \otimes l$ -measure zero [BRE 81], where l is the Lebesgue measure on \mathbb{R}_+ . The process $\{\Lambda_t\}_{t \geq 0}$ defined by

$$\forall t \geq 0, \quad \Lambda_t = \int_0^t \lambda_s ds$$

is \mathcal{H} -adapted with non-decreasing and continuous paths. It is called the \mathcal{H} -compensator of the counting process $\{N_t\}_{t>0}$ and provides the decomposition

$$\forall t \geq 0, \quad N_t = \Lambda_t + M_t. \quad (15.4)$$

For any integrable counting process, the \mathcal{H} -compensator always exists from Doob-Meyer's theorem. Moreover, decomposition (15.4) is unique (up to a set of \mathbb{P} -probability zero). All counting processes in software reliability have a compensator that is absolutely continuous with respect to the Lebesgue measure and therefore have an intensity [BRE 81, Chapitre 2, Theorem 13].

REMARK 15.1.— When the integrability condition (15.3) is not satisfied, the property of \mathcal{H} -martingale only holds for the family of processes $\{N_{t \wedge T_n} - \Lambda_{t \wedge T_n}\}_{t>0}$, $n \geq 1$. The concept of martingale is replaced by that of local martingale.

15.3. Self-exciting point processes and black-box models

The *internal history* of $\{N_t\}_{t \geq 0}$, $\mathcal{H}^N = \{\mathcal{H}_t^N\}_{t \geq 0}$ with $\mathcal{H}_t^N = \sigma(N_s, s \leq t)$ is central in the point process theory. It can be verified that

$$\mathcal{H}_t^N = \sigma(N_t, T_{N_t}, \dots, T_1) \quad \text{and} \quad \mathcal{H}_{T_n}^N = \sigma(T_n, \dots, T_1).$$

The following simplified version of a result due to Jacod [JAC 75] gives an explicit form to the \mathcal{H}^N -intensity.

THEOREM 15.1.— Let $\{N_t\}_{t>0}$ be an integrable counting process. Assume the conditional distribution of the inter-failure time X_{n+1} given $\mathcal{H}_{T_n}^N$ has a density $f_{X_{n+1}|\mathcal{H}_{T_n}^N}$ with $(s, \omega) \mapsto f_{X_{n+1}|\mathcal{H}_{T_n}^N}(s, \omega)$ be $\mathcal{B}(\mathbb{R}_+) \otimes \mathcal{H}_{T_n}^N$ -measurable. Its hazard rate $h_{X_{n+1}|\mathcal{H}_{T_n}^N}$ has form (15.1). Then the process $\{\hat{\lambda}_t\}_{t \geq 0}$ defined by

$$\hat{\lambda}_t = \sum_{n \geq 0} h_{X_{n+1}|\mathcal{H}_{T_n}^N}(t - T_n) \mathbb{I}_{\{T_n < t \leq T_{n+1}\}}$$

is the \mathcal{H}^N -intensity of $\{N_t\}_{t>0}$.

The process $\{\hat{\lambda}_t\}_{t \geq 0}$ is a concatenation of the hazard rate functions of conditional distributions of inter-failure times. A typical path of $\{\hat{\lambda}_t\}_{t>0}$ is illustrated in Figure 15.1. It is clear that any failure model for which the future of the failure process only depends on its past is specified by the sequence of hazard rates $\{h_{X_{n+1}|\mathcal{H}_{T_n}^N}(\cdot)\}_{n \geq 0}$. For instance, the intensity of the celebrated Jelinski-Moranda's model [JEL 72] has the form

$$\hat{\lambda}_t = \Phi(N - N_{t-}).$$

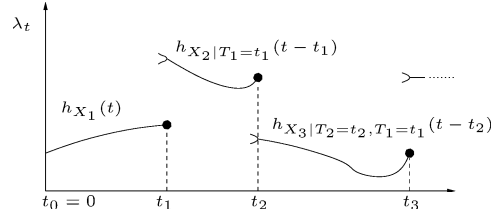


Figure 15.1. A path of $\hat{\lambda}_t$

Its paths are non-increasing and piecewise constant with jumps of size $\Phi > 0$. Here, N must be thought of as the initial number of faults in the software and Φ as a (uniform) factor of the quality of the debugging action after each failure instant. Thus, $\hat{\lambda}_t$ is proportional to the number of residual faults at time t and Φ can also be interpreted as the manifestation rate of any of the N faults.

The concept of “concatenated failure rate” was used in [CHE 97, SIN 99] in connection with “self-exciting point processes” whose definition requires the existence of the limit [SNY 91]

$$\lim_{\Delta t \downarrow 0} \frac{1}{\Delta t} \mathbb{P}\{N_{(t+\Delta t)-} - N_{t-} = 1 \mid \mathcal{H}_{t-}^N\}, \quad (15.5)$$

as well as the “conditional orderliness” condition to hold:

$$\lim_{\Delta t \downarrow 0} \frac{1}{\Delta t} \mathbb{P}\{N_{(t+\Delta t)-} - N_{t-} \geq 1 \mid \mathcal{H}_{t-}^N\} = \lim_{\Delta t \downarrow 0} \frac{1}{\Delta t} \mathbb{P}\{N_{(t+\Delta t)-} - N_{t-} = 1 \mid \mathcal{H}_{t-}^N\}.$$

It can be checked that limit (15.5) is equal to the intensity in definition 15.2:

$$\hat{\lambda}_t = \lim_{\Delta t \downarrow 0} \frac{1}{\Delta t} \mathbb{P}\{N_{(t+\Delta t)-} - N_{t-} = 1 \mid \mathcal{H}_{t-}^N\}.$$

Thus, the intensity expresses the “likelihood” of a system to experience a failure in $[t, t + \Delta t]$, given the number of failures as well as the failure instants just before t . A natural way to introduce growth in reliability is to choose an intensity with non-increasing paths.

The reliability metrics have an explicit form for self-exciting point processes. Recall that the ROCOF, if it exists, is defined by $\mathbb{E}[N_t] = \int_0^t \text{ROCOF}_s ds$. Then, we have:

$$\mathbb{E}[N_t] = \mathbb{E}[\Lambda_t], \quad \text{ROCOF}_t = \mathbb{E}[\hat{\lambda}_t],$$

$$R_t(\tau) = \exp\left(-\int_t^{t+\tau} \hat{\lambda}_s ds\right) = \exp\left(-(\Lambda_{t+\tau} - \Lambda_t)\right),$$

$$\text{MTTF}_t = \mathbb{E}[T_{N_t+1} - t \mid N_t, T_{N_t}, \dots, T_1] = \int_0^{+\infty} R_t(\tau) d\tau.$$

Potentially, any “standard” point process with an \mathcal{H} -stochastic intensity has an intensity with respect to \mathcal{H}^N given by the formula [BRE 81]

$$\hat{\lambda}_t = \mathbb{E}[\lambda_t \mid \mathcal{H}_{t-}^N].$$

A very large collection of software reliability models reported in [MUS 87, XIE 91, LYU 96] have been directly developed using specific forms the sequence of failure rates involved in theorem 15.1. The choice of failure rates is motivated by considering the software system as a single unity. Some of these models are reported in Table 15.1. Note that in general, λ_t only depends on the past of the point process through t, N_t, T_{N_t} . See [LYU 96, ALM 98, PHA 00] for numerical illustrations of black-box models.

Moranda (M)	Generalized order statistics (GOS)	Al Mutairi-Chen-Singpurwalla (ACS)
$\hat{\lambda}_t = \lambda \phi^{N_t-}$ $\phi \in]0, 1[, \lambda \in \mathbb{R}_+$	$\hat{\lambda}_t = (N - N_{t-})\phi(t)$ $N \in \mathbb{N}^*, \phi(\cdot)$ decreasing positive function	$\hat{\lambda}_t = \frac{1}{\frac{t-T_{N_t-}}{k} + \frac{T_{N_t-}}{bN_{t-}}}$ $k \in \mathbb{N}^*, b \in \mathbb{R}_+$
Duane (D)	Goel-Okumoto (GO)	Yamada-Obha-Osaki (YOO)
$\hat{\lambda}_t = \alpha \beta t^{\beta-1}$ $\Lambda_t = \alpha t^\beta$ $\alpha \in \mathbb{R}_+, \beta \in]0, 1[$	$\hat{\lambda}_t = m \phi \exp(-\phi t)$ $\Lambda_t = m(1 - \exp(-\phi t))$ $m, \phi \in \mathbb{R}_+$	$\hat{\lambda}_t = m \phi^2 t \exp(-\phi t)$ $\Lambda_t = m(1 - (1 + \phi t) \exp(-\phi t))$

Table 15.1. *Some black-box models*

Let us briefly comment on the models above (see [GAU 07] for details):

- All the models have a non-increasing intensity which converges to 0, so that the software will become perfect. GO and D illustrate the fact that different convergence rates to 0 of the intensity may be considered in order to take into account the different levels of growth in reliability. A few models have been proposed with a non-zero asymptotic intensity.

- D is a standard model in reliability. Note that a complete statistical analysis of this model is available.

- Model M has been introduced to relax two basic assumptions of model JM, a finite number of initial faults and an uniform factor of quality of any correction. Here, the factor decreases with the number of experienced failures. This reflects the fact that the faults detected and removed in the early phases of testing contribute more to reliability growth than those detected later. λ is the initial failure intensity.

– GOSs cover a wide class of proposed models. N is the initial number of faults in the software and $\hat{\lambda}_t$ is assumed to be proportional to the number of residual faults at time t , so that $\phi(t)$ can be interpreted as the manifestation rate of each residual fault at time t . Note that if we consider a uniform ϕ , we retrieve the model JM.

– GO and YOO belong to the class of bounded NHPP, that is, their compensator converges to the finite value m . Thus, m is interpreted to be the expected number of initial faults. Note that such a property means that the total number of observed failures will be finite with probability 1. Specifically, it has a Poisson distribution with parameter m . Finally, let us mention that if the parameter N in a model GOS is assumed to have a Poisson distribution with parameter m , then the combination of the two random sources gives an NHPP model. For instance, GO is the NHPP version of JM.

– The intensity of YOO has an S -shaped form: it is first increasing and then decreasing. This is supported by the following experimental evidence: the tests are efficient only after a burn-in period. In others words, the ability of engineers to detect bugs in a software increases with time. Note that if $\phi(t) = \phi^2 t / (1 + \phi t)$, then $\hat{\lambda}_t = \phi(t)(m - \Lambda_t)$ and $\phi(t)$ may be interpreted as a factor of the quality of the correction. In this last equality, the case $\phi(t) = \phi$ corresponds to the GO model.

– The ACS model is analyzed from a Bayesian point of view in [ALM 98]. It has very interesting properties. We only list some of them here:

1) The first property is that the mean time to the next failure, MTTF_t , is an increasing function of the time elapsed from the last failure instant $t - T_{N_t}$:

$$k > 1, \quad \text{MTTF}_t = \frac{k}{(k-1)b} \frac{T_{N_t}}{N_t} + \frac{1}{k-1}(t - T_{N_t}).$$

This is supported by the subjective argument that the longer the elapsed time since last failure, the longer the expected time to the next failure.

2) It can be verified that $h_{X_{n+1}|\mathcal{H}_{T_n}}(0) = b(n/T_n)$ which is proportional to the inverse of the mean of the n first inter-failures durations $T_n/n = (\sum_{i=1}^n X_i)/n$. The parameter b appears as a scaling parameter.

3) Let us consider that we have enhanced reliability going from the n st inter-failure time to the $(n+1)$ th inter-failure time if and only if $h_{X_{n+1}|\mathcal{H}_{T_n}^N} \leq h_{X_n|\mathcal{H}_{T_{n-1}}^N}$ on $[0, \min(X_n, X_{n+1})]$. Then this last condition is equivalent to

$$X_n \geq \frac{1}{n-1} \sum_{i=1}^{n-1} X_i.$$

In other words, a growth in reliability implies that the next inter-failure time is greater than the average of all past inter-failure times. Moreover, it can be shown that

$$\frac{\mathbb{E}[X_{n+1}]}{\mathbb{E}[X_n]} \leq 1 \iff b \geq \frac{k}{k-1},$$

that is, on average, we observe a growth in reliability in mean when $b \geq k/(k-1)$.

4) The jump of the intensity at T_n is upward provided that T_n/T_{n-1} satisfies certain conditions.

15.4. White-box models and Markovian arrival processes

The *white-box* (or architecture-based) point of view is an alternative approach in which the architecture of the system is explicitly taken into account. This is advocated for instance in [CHE 80, LIT 79]. The references in the recent reviews on the architecture-based approach [GOS 01, GOK 06] provide a representative sample of the models. We present the main features of Littlewood's model which are common to most works. Note that Cheung's model can be thought of as a discrete time counterpart of Littlewood's one. The following additional papers [SIE 88, KAA 92, OKA 04, LO 05, WAN 06] are concerned with discrete time Markov chain modeling. See [CHE 80, LED 99, GOK 06] for numerical illustrations of white-box models.

In the first step, Littlewood defines an execution model of the software. The basic entity is the standard software engineering concept of *module* as for instance in [CHE 80]. The software architecture is then represented by the *call graph* of the set \mathcal{Y} of the modules. These modules interact by execution control transfers and, at each instant, control lies in one and only one of the modules, which is called the active one. Let us consider the continuous time stochastic process $\{Y_t\}_{t \geq 0}$ where Y_t indicates the active module at time t . $\{Y_t\}_{t \geq 0}$ is assumed to be a homogeneous Markov process on the set $\mathcal{Y} = \{e_1, \dots, e_m\}$ with generator Q . A sequential execution of modules is assumed, so that each state of the execution model corresponds to a module. However, this limitation can be relaxed using an appropriate redesign of the states in \mathcal{Y} (e.g. [TRI 01, WAN 06]).

In the second step, a failure processes is defined. Failure may occur during a control transfer between two modules or during an execution period of any module. When module e_i is active, failures are part of a Poisson process with parameter μ_i . When control is transferred from module e_i to module e_j , a failure may happen with probability $\mu_{i,j}$. Given a sequence of executed modules, all failure processes are assumed to be independent. Generation of failures is illustrated in Figure 15.2.

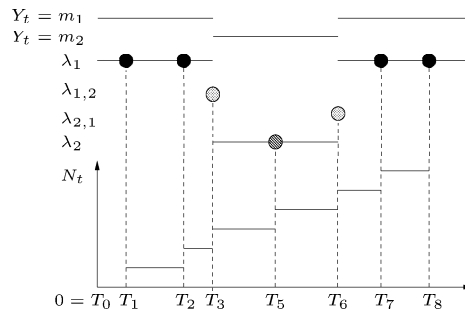


Figure 15.2. Failure arrivals in Littlewood's model with two modules

The execution and failure models are merged into a single model which can then be analyzed. Basically, we are interested in the counting process of failures $\{N_t\}_{t>0}$. Let us make the following comments on Littlewood's model. Each failure does not affect the execution dynamics. The underlying rationale for such an assumption is that the software experiences minor failures from the user point of view, so that the delay to recover a safe state is neglected. The reliability growth is not modeled. When the software is in operation, the reliability growth phenomenon is not very significant and Littlewood's model is thought of as a pessimistic model.

15.4.1. A Markovian arrival model

In this section, we describe a failure model that generalizes Littlewood's [LED 99]. The following assumptions are assumed to hold:

- 1) Two classes of failures in regards to their severity are introduced:
 - a) The first class is composed of failures that provoke an interruption of the delivered service. Such an event is called a *primary failure*. When the software experiences such a failure, the execution returns to a checkpoint or is assumed to be re-initialized from an input component. When component e_i is executed, a primary failure may occur according an exponential distribution with rate λ_i . A transfer of control from state e_i to e_j may also experience a primary failure with probability $\lambda_{i,j}$. After a failure occurs during the execution of component e_i or a transfer of control from e_i , component $e_j \in \mathcal{S}$ is the first component entered with probability $p(i, j)$. Thus, the control flow may be redirected due to the use of error recovery procedures.
 - b) The second class gathers together failures that are not perceived by the user as a non-deliverance of service. In other words, their effects are assumed to be neglectible. Such an event is called *secondary failure*. In this case, the model is exactly like Littlewood's model. A secondary and a primary failure may be simultaneously experienced, but only the primary is taken into account.
- 2) All failure processes are assumed to be independent given the sequence of executed components.
- 3) The delays to recover a safe state are assumed to be negligible.

The third assumption is supported by the fact that the breakdown periods are often much shorter than the inter-failure times in the operating phase of the software lifecycle. This assumption is relaxed in [LED 99]. The reliability growth is still not taken into account. In contrast to Littlewood's model, the failure may affect the execution dynamics of the system. Let us denote the entered component just after the n th failure instant T_n by $Y_{T_n}^*$. Then $\{Y_{T_n}^*\}_{n \geq 0}$ is a Markov chain from the assumptions and the occupation times of each component e_i is exponentially distributed with parameter $-Q(i, i) + \lambda_i + \mu_i$. We define the following jump Markov process $\{Y_t^*\}_{t \geq 0}$ as

$$Y_t^* = Y_{T_n}^* \quad \text{when } T_n \leq t < T_{n+1}.$$

Now, it is easily seen that the bivariate process $\{(N_t, Y_t^*)\}_{t>0}$, where $\{N_t\}_{t>0}$ is the counting process of failures, is a jump Markov process with state space $\mathbb{N} \times \mathcal{Y}$. Its generator has the following form using the lexicographic order on $\mathbb{N} \times \mathcal{Y}$:

$$\begin{pmatrix} D_0 & D_1 & \mathbf{0} & \cdots \\ \mathbf{0} & D_0 & D_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix}$$

with $m \times m$ -matrices D_0, D_1 defined by: for $i, j = 1, \dots, m$,

$$D_0(i, j) = Q(i, j)(1 - \lambda_{i,j})(1 - \mu_{i,j}) \quad i \neq j, \quad D_0(i, i) = Q(i, i) - \lambda_i - \mu_i,$$

$$D_1(i, j) = Q(i, j)[1 - \lambda_{i,j}]\mu_{i,j} + [\lambda_i + \sum_{k \neq i} Q(i, k)\lambda_{i,k}]p(i, j).$$

The generator of the Markov process $\{Y_t^*\}_{t>0}$ is

$$Q^* = D_0 + D_1$$

Q^* being assumed to be irreducible so that $\mathbb{P}\{N_\infty = +\infty\} = 1$. Y_0^*, Y_0 have the same probability distribution that is denoted by the vector $\alpha = (\alpha(i))_{i=1}^m$. It is interpreted as the usage profile of the various components. Littlewood's model corresponds to matrices D_0, D_1 with $\lambda_i = \lambda_{i,j} = 0$, so that $Q^* = Q$. The simplest model for which a failure affects the execution process is the PH-renewal process. Let us only consider primary failures in the modules, i.e. $\mu_i = 0, \lambda_{i,j} = \mu_{i,j} = 0, i, j = 1, \dots, m$. After a failure has been experienced, the control is redirected to the component e_j with probability α_j , independently of where the failure occurred: $i, j = 1, \dots, m, p(i, j) = \alpha_j$. We obtain with $\underline{\lambda} = (\lambda_i)_{i=1}^m$,

$$D_0 = Q - \text{diag}(\lambda_i), \quad D_1 = \underline{\lambda}^1 \alpha = -D_0 \mathbf{1}^\top \alpha, \quad Q^* = D_0 - D_0 \mathbf{1}^\top \alpha.$$

As in the self-exciting context, explicit formulae for the various reliability metrics can be given. Let $\{X_n\}_{n>0}$ be the inter-failure times. Let us denote the exponential matrix function $\exp(D_0 t)$ by $f_0(t)$. Then the \mathcal{H}^N -intensity $\{\hat{\lambda}_t\}_{t \geq 0}$ of $\{N_t\}_{t \geq 0}$ is

$$\hat{\lambda}_t = \frac{\alpha f_0(X_1) D_1 \cdots f_0(X_{N_t}) D_1 f_0(t - T_{N_t}) D_1 \mathbf{1}^1}{\alpha f_0(X_1) D_1 \cdots f_0(X_{N_t}) D_1 f_0(t - T_{N_t}) \mathbf{1}^\top}$$

and the main reliability metrics have the following form

$$\begin{aligned} R_t(\tau) &= \frac{\alpha f_0(X_1) D_1 \cdots f_0(X_{N_t}) D_1 f_0(t - T_{N_t}) f_0(\tau) \mathbf{1}^1}{\alpha f_0(X_1) D_1 \cdots f_0(X_{N_t}) D_1 f_0(t - T_{N_t}) \mathbf{1}^\top}, \\ \text{MTTF}_t &= \frac{\alpha f_0(X_1) D_1 \cdots f_0(X_{N_t}) D_1 f_0(t - T_{N_t}) (-D_0)^{-1} \mathbf{1}^1}{\alpha f_0(X_1) D_1 \cdots f_0(X_{N_t}) D_1 f_0(t - T_{N_t}) \mathbf{1}^\top}, \\ \mathbb{E}[N_t] &= \int_0^t \alpha \exp(Q^* u) du D_1 \mathbf{1}^\top, \quad \text{ROCOF}_t = \alpha \exp(Q^* t) D_1 \mathbf{1}^\top. \end{aligned}$$

The distribution function of random variable N_t also has an explicit form [LED 99]. The uniformization method can be used to calculate the exponential matrices [STE 94].

15.4.2. Parameter estimation

The multidimensional parameter $\theta = \{D_0(i, j), D_1(i, j), i, j = 1, \dots, m\}$ must be estimated. The likelihood function for observations $t_1 < \dots < t_k$ is

$$\theta \mapsto \mathcal{L}(\theta; t_1, \dots, t_k) = \alpha f_0(t_1) D_1 f_0(t_2 - t_1) D_1 \cdots f_0(t_k - t_{k-1}) D_1 \mathbf{1}^1. \quad (15.6)$$

This function is highly non-linear, so that a standard numerical procedure fails to optimize it when m is large. Estimating θ can be thought of as an estimation problem in a missing-data context. Indeed, θ is associated with the Markov process $\{(N_t, Y_t^*)\}_{t \geq 0}$ and must be estimated from observations of its first component $\{N_t\}_{t \geq 0}$. This suggests the use of the expectation-maximization (EM) methodology. Vector α is assumed to be known. In the following, t denotes the last failure time t_k .

Let us introduce the *complete data likelihood*, that is, the likelihood associated with the observation of the bivariate Markov process $\{(N_t, Y_t^*)\}_{t \geq 0}$ over $[0, t]$ [KLE 03]:

$$\begin{aligned} L_t(\theta; N, Y^*) \\ = \prod_{i=1}^m \alpha(i)^{\mathbf{1}_{\{Y_0^* = e_i\}}} \prod_{i=1}^m e^{D_0(i, i) \mathcal{O}_t^{(i)}} \prod_{i, j=1, j \neq i}^m D_0(i, j)^{\mathcal{L}_t^{0, ij}} \prod_{i, j=1}^m D_1(i, j)^{\mathcal{L}_t^{1, ij}} \end{aligned}$$

where

$$\begin{aligned} \mathcal{L}_t^{1, ij} &= \sum_{0 < s \leq t} \Delta N_s \mathbf{1}_{\{Y_{s-}^* = e_i, Y_s^* = e_j\}} = \int_0^t \mathbf{1}_{\{Y_{s-}^* = e_i, Y_s^* = e_j\}} dN_s, \\ j \neq i, \mathcal{L}_t^{0, ij} &= \sum_{0 < s \leq t} (1 - \Delta N_s) \mathbf{1}_{\{Y_{s-}^* = e_i, Y_s^* = e_j\}}, \quad \mathcal{O}_t^{(i)} = \int_0^t \mathbf{1}_{\{Y_{s-}^* = e_i\}} ds \end{aligned}$$

and ΔN_s is the jump size $N_s - N_{s-}$ of the counting process at time s .

For $i \neq j$, $\mathcal{L}_t^{1, ij}$ is the number of failures with a transfer of control from e_i to e_j up to time t . $\mathcal{L}_t^{1, ii}$ is the number of observed failures at time t with no transition of $\{Y_t^*\}_{t \geq 0}$ from state e_i . For $i \neq j$, $\mathcal{L}_t^{0, ij}$ is the number of control transfers from e_i to e_j with no failure. The last statistics is the occupation time of state e_i by $\{Y_t^*\}_{t \geq 0}$ on $[0, t]$. Note that the complete data log-likelihood has the form

$$\begin{aligned} \log L_t(\theta; N, Y^*) &= \sum_{i, j=1, j \neq i}^m \log D_0(i, j) \mathcal{L}_t^{0, ij} + \sum_{i, j=1}^m \log D_1(i, j) \mathcal{L}_t^{1, ij} \\ &\quad + \sum_{i=1}^m D_0(i, i) \mathcal{O}_t^{(i)} + K \end{aligned}$$

where K does not depend on θ .

Let \mathbb{P}_θ denotes the probability model under θ . A sequence $\{\theta_l\}_{l \in \mathbb{N}}$ is obtained from the iteration of the following maximization procedure:

$$\theta_{l+1} = \operatorname{argmax}_{\theta^*} \mathbb{E}_{\theta_l} [\log L_t(\theta^*; N, Y^*) \mid \mathcal{H}_t^N].$$

Using the Lagrange multipliers method, $\theta_{l+1} = \{D_0^{(l+1)}(i, j), D_1^{(l+1)}(i, j), i, j = 1, \dots, m\}$ is shown to be

$$D_1^{(l+1)}(i, j) = \frac{\mathbb{E}_{\theta_l}[\mathcal{L}_t^{1,ij} \mid \mathcal{H}_t^N]}{\mathbb{E}_{\theta_l}[\mathcal{O}_t^{(i)} \mid \mathcal{H}_t^N]}, \quad D_0^{(l+1)}(i, j) = \frac{\mathbb{E}_{\theta_l}[\mathcal{L}_t^{0,ij} \mid \mathcal{H}_t^N]}{\mathbb{E}_{\theta_l}[\mathcal{O}_t^{(i)} \mid \mathcal{H}_t^N]} \quad i \neq j. \quad (15.7)$$

The crucial fact is that the sequence $\{\theta_l\}_{l \in \mathbb{N}}$ makes the likelihood function (15.6) non-decreasing, i.e.

$$L(\theta_{l+1}; t_1, \dots, t_k) \geq L(\theta_l; t_1, \dots, t_k).$$

It can be shown that it is actually an equality if and only if $\theta_{l+1} = \theta_l$ under natural conditions [CAP 05]. Note that the zero coefficients of D_0, D_1 are preserved by the procedure.

The EM-algorithm consists of selecting an initial value θ_0 and iterates the calculation of (15.7) as long as some stopping criterion is not met. θ_0 may be evaluated from empirical methods based on data gathered in the early phases of the software lifecycle (validation phase, integration tests, etc.) [GOS 01]. Thus, the estimate θ_0 is not representative of the operating phase and must be considered as an *a priori* estimate of θ . In this context, the EM procedure must be thought of as a recalibration method of θ_0 in view of failure data observed in the operating phase of the software lifecycle. The conditional expectations in (15.7) may be calculated using the well-known forward-backward or Baum-Welch technique. The starting point is to exchange the order of integration in (15.7) to calculate the average in time, for $s \leq t$, of:

$$\mathbb{P}\{Y_s = e_i \mid \mathcal{H}_t^N\}, \quad \mathbb{P}\{\Delta N_s = p, Y_{s-} = e_i, Y_s = e_j \mid \mathcal{H}_t^N\} \quad p = 0, 1$$

recalling that $t = t_k$. These conditional probabilities are called *smoothers* and are calculated from a forward and a backward recursive formula [KLE 03, and the references therein].

For hidden Markov chains, an alternative approach based on a direct recursive calculation of the conditional expectations is proposed by Elliott [ELL 95]. The material for implementing this approach to the Markovian model introduced here, are given in [LED 07a]. In contrast to the Baum-Welch technique, only a single pass through the data is needed. Thus, the memory requirement is independent of the number of observations. Moreover, an on-line estimation of θ is allowed. The disadvantage of the

filter-based method is that each conditional expectation requires one recursive formula to be implemented. A detailed discussion on the respective properties of the two approaches may be found in [CAP 05] for the discrete time case and in [LED 06] for the present context. Now, we turn back to the recursive equations derived in [LED 07a] using a change of probability measure. Indeed, there exists a probability measure \mathbb{P}_0 under which $\{N_t\}_{t \geq 0}$ is the counting process of a Poisson process with intensity 1 and $\{Y_t^*\}_{t > 0}$ is a Markov chain with generator $D_0 + D_1$. Let e_i be the i th vector of the canonical basis of \mathbb{R}^m so that

$$\mathbb{1}_{\{Y_t^* = e_i\}} = Y_t^*(i).$$

THEOREM 15.2.— [LED 07a, Th 2] *Let L_t be the likelihood ratio over the interval $[0, t]$ associated with the counting process $\{N_t\}_{t \geq 0}$ of intensity $\lambda_t = Y_{s-}^* D_1 \mathbf{1}^\top$:*

$$L_t = \prod_{0 < s \leq t} \lambda_s^{\Delta N_s} \exp \left(\int_0^t (1 - \lambda_s) ds \right)$$

Set $\sigma(Z_t) = \mathbb{E}_0[Z_t L_t \mid \mathcal{H}_t^N]$ for any \mathcal{H}^{N, Y^} -adapted integrable process $\{Z_t\}_{t > 0}$. Let $\{n_t\}_{t > 0}$ be the process defined by $n_t = N_t - t$. We have for any $t \geq 0$,*

$$\begin{aligned} \sigma(Y_t^*) &= \mathbb{E}[Y_0^*] + \int_0^t \sigma(Y_{s-}^*) Q ds + \int_0^t \sigma(Y_{s-}^*) (D_1 - I) dn_s, \\ \sigma(\mathcal{O}_t^{(i)} Y_t^*) &= \int_0^t [\sigma(\mathcal{O}_{s-}^{(i)} Y_{s-}^*) Q + \sigma(Y_{s-}^*)(i) e_i] ds \\ &\quad + \int_0^t \sigma(\mathcal{O}_{s-}^{(i)} Y_{s-}^*) (D_1 - I) dn_s, \\ \sigma(\mathcal{L}_t^{0,ij} Y_t^*) &= \int_0^t [\sigma(\mathcal{L}_{s-}^{0,ij} Y_{s-}^*) Q + D_0(i, j) \sigma(Y_{s-}^*)(i) e_j] ds \\ &\quad + \int_0^t \sigma(\mathcal{L}_{s-}^{0,ij} Y_{s-}^*) (D_1 - I) dn_s, \\ \sigma(\mathcal{L}_t^{1,ij} Y_t^*) &= \int_0^t [\sigma(\mathcal{L}_{s-}^{1,ij} Y_{s-}^*) Q + D_1(i, j) \sigma(Y_{s-}^*)(i) e_j] ds \\ &\quad + \int_0^t [\sigma(\mathcal{L}_{s-}^{1,ij} Y_{s-}^*) (D_1 - I) + D_1(i, j) \sigma(Y_{s-}^*)(i) e_j] dn_s. \end{aligned}$$

The conditional expectations of $\mathcal{O}_t^{(i)}, \mathcal{L}_t^{1,ij}, \mathcal{L}_t^{0,ij}$ are $\sigma(\mathcal{O}_t^{(i)}) = \sigma(\mathcal{O}_t^{(i)} Y_t^*) \mathbf{1}^\top$, $\sigma(\mathcal{L}_t^{1,ij}) = \sigma(\mathcal{L}_t^{1,ij} Y_t^*) \mathbf{1}^\top$, $\sigma(\mathcal{L}_t^{0,ij}) = \sigma(\mathcal{L}_t^{0,ij} Y_t^*) \mathbf{1}^\top$. Finally, the conditional expectations under the original probability \mathbb{P} , are obtained as follows:

$$\mathbb{E}[\mathcal{O}_t^{(i)} \mid \mathcal{H}_t^N] = \frac{\sigma(\mathcal{O}_t^{(i)})}{\sigma(1)}, \quad \mathbb{E}[\mathcal{L}_t^{0,ij} \mid \mathcal{H}_t^N] = \frac{\sigma(\mathcal{L}_t^{0,ij})}{\sigma(1)}, \quad \mathbb{E}[\mathcal{L}_t^{1,ij} \mid \mathcal{H}_t^N] = \frac{\sigma(\mathcal{L}_t^{1,ij})}{\sigma(1)}.$$

This set of stochastic differential equations provides the following algorithm [LED 07a].

RECURSIVE ALGORITHM.—

$$\begin{aligned}
 f_0(x) &= \exp(D_0 x), f_1(x) = D_1 f_0(x); \Delta t_l = t_l - t_{l-1}, l = 1, \dots, k \text{ with } t_0 = 0. \\
 \text{for } l &= 1, \dots, k \\
 \sigma(X_{t_l}) &= \sigma(Y_{t_{l-1}}^*) f_1(\Delta t_l) \\
 \sigma(\mathcal{L}_{t_l}^{0,ij} Y_{t_l}^*) &= \sigma(\mathcal{L}_{t_{l-1}}^{0,ij} Y_{t_{l-1}}^*) f_1(\Delta t_l) + \sigma(Y_{t_{l-1}}^*) \int_{t_{l-1}}^{t_l} f_0(s - t_{l-1}) e_i^\top D_0(i, j) e_j f_1(t_l - s) ds \\
 \sigma(\mathcal{L}_{t_l}^{1,ij} Y_{t_l}^*) &= f_1(\Delta t_l) \sigma(\mathcal{L}_{t_{l-1}}^{1,ij} Y_{t_{l-1}}^*) + e_j D_1(j, i) e_i^\top f_0(\Delta t_l) \sigma(Y_{t_{l-1}}^*) \\
 \sigma(\mathcal{O}_{t_l}^{(i)} Y_{t_l}^*) &= f_1(\Delta t_l) \sigma(\mathcal{O}_{t_{l-1}}^{(i)} Y_{t_{l-1}}^*) + \int_{t_{l-1}}^{t_l} f_1(t_l - s) e_i e_i^\top f_0(s - t_{l-1}) ds \sigma(Y_{t_{l-1}}^*) \\
 \text{Comment.} & \text{ The factor } \exp(\Delta t_l) \text{ is omitted in the equations above, because the estimates (15.7) of } D_0, D_1 \text{ only require the knowledge of the filters up to a constant.}
 \end{aligned}$$

In general, the EM-algorithm converges to a local maxima of the likelihood function (15.6). Its main drawback is its slow convergence. See [WU 83, CAP 05] for details. Experiments show that the procedure is robust. In the specific context of the architecture-based software reliability modeling, the use of EM-algorithm was suggested in [LED 07a]. See [LED 06] for a detailed discussion in the context of discrete/continuous time software reliability modeling.

15.4.3. Reliability growth

The previous models do not take into account the expected reliability growth of a software. As in [LIT 75, LIT 79], we investigate the limit model when the failure parameters become much smaller than the parameters that drive exchanges of control. In this context, Littlewood claimed that $\{N_t\}_{t \geq 0}$ converges in distribution to a homogeneous Poisson process with intensity

$$\lambda_{eq} = \sum_{i=1}^m \pi(i) \left[\mu_i + \sum_{j \neq i} Q(i, j) \mu_{i,j} \right], \quad (15.8)$$

where $(\pi(i))_{i=1, \dots, m}$ is the invariant distribution of the irreducible generator Q . The next theorem states that a similar limit result holds for a version of the model of section 15.4.1 in which $\{Y_t\}_{t > 0}$ is assumed to be a non-homogeneous Markov process with a family of measurable generators $\{Q(t)\}_{t > 0}$ satisfying

$$\sup_t \|Q(t)\|_1 = \sup_t \max_i \left(\sum_{j=1}^m |Q(t)(i, j)| \right) < \infty.$$

A direct way to introduce the problem is to multiply the failure parameters by a small parameter $\varepsilon > 0$. As a result, we obtain a Markovian model with matrices $\{D_0^{(\varepsilon)}(t), D_1^{(\varepsilon)}(t)\}_{t \geq 0}$ of the form

$$D_0^{(\varepsilon)}(t) = Q(t) + \varepsilon B_0(t) + \varepsilon^2 L_0(t) \quad \text{and} \quad D_1^{(\varepsilon)}(t) = \varepsilon B_1(t) + \varepsilon^2 L_1(t),$$

where $B_0(t), L_0(t), B_1(t)$ and $L_1(t)$ are matrices of uniformly bounded measurable functions such that

$$(B_0(t) + B_1(t))\mathbf{1}^\top = (L_0(t) + L_1(t))\mathbf{1}^\top = \mathbf{0}.$$

The Markov process $\{Y_t^*\}_{t \geq 0}$ has generator $\{Q^{*,\varepsilon}(t)\}_{t \geq 0}$ with

$$Q^{*,\varepsilon}(t) = D_0^{(\varepsilon)}(t) + D_1^{(\varepsilon)}(t) = Q(t) + \varepsilon R_0(t) + \varepsilon^2 R_1(t).$$

The main assumption is on the rate of convergence of $Q(t)$ to some irreducible generator Q with stationary distribution π :

$$\exists \beta > 1, \quad \lim_{t \rightarrow +\infty} (2t)^\beta \|Q(t) - Q\|_1 = 0. \quad (15.9)$$

Such a condition implies that $B_1(t)$ converges to a non-negative matrix B_1 . Then, we investigate the asymptotic distribution of the counting process $\{N_t^{(\varepsilon)}\}_{t \geq 0}$ defined by

$$\forall t \geq 0, \quad N_t^{(\varepsilon)} = N_{t/\varepsilon}.$$

Let us introduce the distance in variation between the probability distributions $\mathcal{L}(N_{\mathbb{T}}^{(\varepsilon)})$ and $\mathcal{L}(\Pi_{\mathbb{T}})$ of random vectors $N_{\mathbb{T}}^{(\varepsilon)} = (N_{t_1}^{(\varepsilon)}, \dots, N_{t_n}^{(\varepsilon)})$ and $P_{\mathbb{T}} = (P_{t_1}, \dots, P_{t_n})$:

$$d_{\text{TV}}(\mathcal{L}(N_{\mathbb{T}}^{(\varepsilon)}), \mathcal{L}(P_{\mathbb{T}})) = \sup_{B \subset \mathbb{N}^n} |\mathbb{P}\{N_{\mathbb{T}}^{(\varepsilon)} \in B\} - \mathbb{P}\{P_{\mathbb{T}} \in B\}|.$$

THEOREM 15.3.– [LED 07b, Th 4] *Let $\{P_t\}_{t \geq 0}$ be a Poisson process with intensity*

$$\lambda_{eq} = \pi B_1 \mathbf{1}^\top = \sum_{i=1}^m \pi(i) [\mu_i + \lambda_i + \sum_{j \neq i} Q(i, j)(\lambda_{i,j} + \mu_{i,j})]. \quad (15.10)$$

Under condition (15.9), for any $T > 0$ there exists a constant C_T such that

$$d_{\text{TV}}(\mathcal{L}(N_{\mathbb{T}}^{(\varepsilon)}), \mathcal{L}(P_{\mathbb{T}})) \leq C_T \varepsilon.$$

For Littlewood's model, (15.10) reduces to (15.8). The result provides a rate at which the convergence takes place. An important fact is that the order of the convergence rate is optimal (see [LED 07b, Rem. 5]).

REMARK 15.2.– In fact, the convergence in variation takes place in the Skorokhod space [LED 07b, Th. 4].

REMARK 15.3.– In Littlewood's model, when $\{Y_t\}_{t \geq 0}$ is an irreducible semi-Markovian process, Littlewood claimed that $\{N_t\}_{t \geq 0}$ is still asymptotically a Poisson process [LIT 79]. This fact is justified in [LED 03b, Rem. 2].

15.5. Bibliography

- [ALM 98] AL-MUTAIRI D., CHEN Y., SINGPURWALLA N. D., “An adaptative concatenated failure rate model for software reliability”, *J. Amer. Statist. Ass.*, vol. 93, num. 443, p. 1150–1163, 1998.
- [ASC 84] ASCHER H., FEINGOLD H., *Repairable Systems Reliability: Modeling, Inference, Misconceptions and their Causes*, Marcel Dekker, 1984.
- [BRE 81] BREMAUD P., *Point Processes and Queues*, Springer, 1981.
- [CAP 05] CAPPÉ O., MOULINES E., RYDÉN T., *Inference in Hidden Markov Models*, Springer, 2005.
- [CHE 80] CHEUNG R. C., “A user-oriented software reliability model”, *IEEE Trans. Software Eng.*, vol. 6, p. 118–125, 1980.
- [CHE 97] CHEN Y., SINGPURWALLA N. D., “Unification of software reliability models by self-exciting point processes”, *Adv. in Appl. Probab.*, vol. 29, p. 337–352, 1997.
- [ELL 95] ELLIOTT R. J., AGGOUN L., MOORE J. B., *Hidden Markov Models*, Springer, 1995.
- [GAU 90] GAUDOIN O., “Outils statistiques pour l’évaluation de la fiabilité du logiciel”, PhD thesis, Université Joseph Fourier–Grenoble I, 1990.
- [GAU 07] GAUDOIN O., LEDOUX J., *Modélisation aléatoire en fiabilité des logiciels*, Hermès, 2007.
- [GOK 06] GOKHALE S. S., TRIVEDI K. S., “Analytical models for architecture-based software reliability prediction: a unification framework”, *IEEE Trans. Reliab.*, vol. 55, p. 578–590, 2006.
- [GOS 01] GOSEVA-POPSTOJANOVA K., TRIVEDI K. S., “Architecture-based approach to reliability assessment of software systems”, *Perfor. Evaluation*, vol. 45, p. 179–204, 2001.
- [JAC 75] JACOD J., “Multivariate point processes: predictable projection, Radon-Nikodym derivatives, representation of martingales”, *Z. Wahrsch. Verw. Gebiete*, vol. 31, p. 235–253, 1975.
- [JEL 72] JELINSKI Z., MORANDA P. B., “Software reliability research”, In FREIBERGER W., Ed., *Statistical computer performance evaluation*, p. 465–484, Academic Press, 1972.
- [KAÂ 92] KAÂNICHE M., KANOUN K., “The discrete time hyperexponential model for software reliability growth evaluation”, In *Int. Symp. on Software Reliability*, p. 64–75, 1992.
- [KLE 03] KLEMM A., LINDEMANN C., LOHMANN M., “Modeling IP traffic using the batch Markovian arrival process”, *Performance Evaluation*, vol. 54, p. 149–173, 2003.
- [LED 99] LEDOUX J., “Availability modeling of modular software”, *IEEE Trans. Reliab.*, vol. 48, p. 159–168, 1999.
- [LED 03a] LEDOUX J., “Software reliability modeling”, In PHAM H., Eds., *Handbook of Reliability Engineering*, p. 213–234, Springer, 2003.

- [LED 03b] LEDOUX J., “On the asymptotic analysis of Littlewood’s reliability model for modular software”, In LINDQVIST B. H., DOKSUM K. A., Eds., *Mathematical and Statistical Methods in Reliability*, p. 521–536, World Scientific, 2003.
- [LED 06] LEDOUX J., “EM-algorithm in for software modeling”, In *Int. Conf. on Degrad., Damage, Fatigue and Accel. Life Models in Reliab. Testing (ALT’06)*, p. 278–286, 2006.
- [LED 07a] LEDOUX J., “Filtering and the EM-algorithm for the Markovian arrival process”, *Communications in Statistics–Theory and Methods*, vol. 36, num. 14, 2007.
- [LED 07b] LEDOUX J., “Strong convergence of a class of non-homogeneous Markovian arrival process to a Poisson process”, To appear in *Statistics and Probability Letters*, 2007.
- [LIT 75] LITTLEWOOD B., “A Reliability model for systems with Markov structure”, *J. Roy. Statist. Soc. Ser. C*, vol. 24, p. 172–177, 1975.
- [LIT 79] LITTLEWOOD B., “Software reliability model for modular program structure”, *IEEE Trans. Reliab.*, vol. 28, p. 241–246, 1979.
- [LO 05] LO J.-H., HUANG C.-Y., CHEN I.-Y., KUO S.-Y., LYU M. R., “Reliability assessment and sensitivity analysis of software reliability growth modeling based on software module structure”, *The Journal of Systems and Software*, vol. 76, p. 3–13, 2005.
- [LYU 96] LYU M. R., Ed., *Handbook of Software Reliability Engineering*, IEEE Computer Society Press and McGraw-Hill Book Company, 1996. see <http://www.cse.cuhk.edu.hk/lyu/book/reliability/>
- [MUS 87] MUSA J. D., IANNINO A., OKUMOTO K., *Software Reliability: Measurement, Prediction, Application.*, McGraw-Hill International Editions, 1987.
- [OKA 04] OKAMURA H., KUROKI S., DOHI T., OSAKI S., “A reliability growth model for modular software”, *Electronics and Comm. in Japan*, vol. 87, p. 905–914, 2004.
- [PHA 00] PHAM H., *Software Reliability*, Springer, 2000.
- [SIE 88] SIEGRIST K., “Reliability of systems with Markov transfer of control, II.”, *IEEE Trans. Software Eng.*, vol. 14, p. 1478–1480, 1988.
- [SIN 99] SINGPURWALLA N. D., WILSON S., *Statistical Methods in Software Engineering*, Springer, 1999.
- [SNY 91] SNYDER D. L., MILLER M. I., *Random Point Processes in Time and Space*, Springer, 1991.
- [STE 94] STEWART W. J., *Introduction to the Numerical Solution of Markov chains*, Princeton University, 1994.
- [TRI 01] TRIVEDI K. S., *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, John Wiley and Sons, 2nd Edition, 2001.
- [WAN 06] WANG W.-L., PAN D., , CHEN M.-H., “Architecture-based software reliability modeling”, *J. Systems and Software*, vol. 79, p. 132–146, 2006.
- [WU 83] WU C. F. J., “On the convergence properties of the EM algorithm”, *Ann. Statist.*, vol. 11, p. 95–103, 1983.
- [XIE 91] XIE M., *Software Reliability Modeling*, World Scientific Publishing, 1991.